Int J Life Cycle Assess (2012) 17:774–786
DOI 10.1007/s11367-012-0418-7

Systematic disaggregation: a hybrid LCI computation algorithm enhancing interpretation phase in LCA

Guillaume Bourgault · Pascal Lesage · Réjean Samson

Received: 5 April 2011 / Accepted: 19 March 2012 / Published online: 24 April 2012

Abstract

Purpose Before the advent of large databases, practitioners often lacked data for calculating life cycle inventories, but the actual computation was a straightforward task. Now that databases represent supply chains including feedback loops and several thousand unit processes and emissions, more formalized calculation methods are necessary. Two methods are widely used: sequential method and matrix inversion. They both exhibit different advantages and drawbacks. The present paper proposes a hybrid algorithm combining the advantages of both methods while minimizing their inconveniences.

Methods Sequential algorithm requires a form of cutoff criteria, as the supply chains are of infinite length in the presence of feedback loops. The proposed implementation allows the detailing of individual paths until their upstream contribution falls below a user-defined disaggregation criteria, while also allowing the total impact scores of all paths to be stored and considered. The output is then structured to facilitate consultation and re-aggregation, enhancing the work of practitioners in the interpretation phase of LCA. The algorithm is a variation on structural path analysis and accumulative structural path analysis. It is computationally efficient and uses a reporting threshold criterion based on multiple impact categories.

Results Although the algorithm leads to a more voluminous inventory than matrix inversion, it produces detailed, useful information on the particular instances of processes responsible for the impacts. An average laptop can compute the results within seconds. This algorithm has the potential to improve the interpretation phase of LCA. More specifically, selective replacement of values (characterization factors, input from technosphere, or emission intensities) in parts of the process tree can be applied without affecting the rest of the system.

Conclusions LCA software would benefit from the inclusion of the algorithm presented in this paper. It produces additional information on the structure of the supply chain and the impacts of its constituents, which would be available for a more in-depth interpretation by practitioners. Its potential for understanding the propagation of uncertainty and acceleration of Monte Carlo assessment should also be investigated.

Keywords Disaggregation · Interpretation phase · LCI algorithm · Matrix computation · Structural path analysis

1 Introduction

Life cycle assessment (LCA) requires the modeling of a large number of intermediary product exchanges between clients and suppliers. A small team cannot gather the large body of information for each LCA. In practice, generic databases representing the background web of exchanges are used, as the LCA practitioners focus on data collection of foreground processes. Databases that contain information on processes that are at the unit process level (i.e., are not pre-aggregated) usually contain feedback loops, reflecting the fact that supply chains are, in theory, infinite. Special methods are

Responsible editor: Reinout Heijungs

G. Bourgault (✉) · P. Lesage · R. Samson
CIRAIG, Chemical Engineering Department,
École Polytechnique de Montréal,
P.O. Box 6079, Montreal, QC H3C 3A7, Canada
e-mail: guillaume.bourgault@polymtl.ca

required to compute the impacts of an infinite supply chain. Otherwise, the number of mathematical operations grows exponentially, and the amount of information on emissions rapidly becomes unmanageable.

Two types of algorithm are available to solve the linear system to reach the inventory of the whole supply chain of a reference flow: those derived from the matrix inversion and those derived from the sequential method (described in Sections 2.1 and 2.2). The latter gives access to a disaggregated inventory. However, it can lead to truncation errors, whereby supply chains upstream of selected nodes are cutoff to reduce computation time or, conversely, to unacceptably long computation time if one seeks to minimize truncation errors. Matrix inversion, on the other hand, is a fairly straightforward manner to provide an exact inventory for infinite supply chains (i.e., to avoid truncation errors), as those found in our highly intricate economy. However, this approach produces a partially aggregated inventory, which impedes an in-depth analysis of contributions.

A number of papers have presented structural path analysis (SPA; related to the sequential method) as a means to answer specific research questions in the context of input–output tables (Defourny and Thorbecke 1984; Lenzen 2002; Peters and Hertwich 2006). However, we will show in Section 2.2 that they are not completely adapted to process LCA. The description of the algorithm used in those papers is usually formalistic and focused on a specific research questions.

We believe algorithms based on the sequential approach are under-utilized in LCA, both in the day-to-day practice and to answer research questions. They have been presented in a very formalistic way that do not make the approach user-friendly and do not reveal their full potential.

This paper presents a hybrid of the sequential and matrix inversion algorithms. The new algorithm, which we name systematic disaggregation, is described with full detail and designed in the most general manner to meet the changing needs of different LCA studies, as well as a research tool. The algorithm is inspired by the previously published papers on the subject, particularly Suh and Heijungs (2007) and Suh (2004), and works on the same basic principles. Its originality lies in the structure of the output, allowing greater flexibility in the interpretation phase. The performance of the algorithm is analyzed. Finally, specific examples are shown to demonstrate how the disaggregated output of the algorithm enhances the interpretation phase of LCA.

2 Methodology

2.1 Matrix inversion

A very fast and easily implemented way to calculate the total impact of a system is the matrix inversion approach (Heijungs and Suh 2002). An outline of essential variables of this algorithm can be found at Table 1.

How inverting a matrix accounts an infinite supply chain can best be demonstrated using power series expansion. It has been shown that under certain conditions (Suh and Heijungs 2007), the following relation stands:

$$\begin{aligned}\mathbf{A}^{-1} &= \mathbf{I} + (\mathbf{I}-\mathbf{A}) + (\mathbf{I}-\mathbf{A})^2 + (\mathbf{1}-\mathbf{A})^3 + \ldots \\ &= \mathbf{I} + \mathbf{Z} + \mathbf{Z}^2 + \mathbf{Z}^3 + \ldots\end{aligned}$$

The basic $\mathbf{Z}$ matrix is defined as $\mathbf{Z}=\mathbf{I}-\mathbf{A}$. Each $\mathbf{Z}^n$ matrix represents the contribution of each tier (other equivalent expressions found in the literature: "production layer," "round of the supply chain") to the scaling vector. Since the infinite sum converges to the inverse of the matrix, using the matrix inversion is analogous to integrating over a supply chain of infinite length.

Matrix inversion produces an aggregated scaling vector. It represents the total of the direct and indirect output each unit process has to produce to supply the reference flows, i.e., the sum of the production of each individual instance, or occurrence, of each unit process. The contributions of each of those instances are automatically summed together: Individual contributions are not accessible. This degree of aggregation is carried on along the rest of the calculation and leads to the total impact scores of the system, without further detail. If only this information is needed, this algorithm fulfills all expectations.

Some more details can be obtained by transforming the scaling vector into a square matrix, where the diagonal corresponds to the scaling vector:

$\mathbf{S}(i,j) = \mathbf{s}(i)$ if $i = j$, zero otherwise

By substituting this matrix to the scaling vector in the algorithm described in Table 1, an $m \times n$ inventory matrix and a $k \times n$ impact score matrix are obtained. The scores have been disaggregated around n poles, corresponding to the unit processes defined in the database. This allows discriminating which unit processes of the life cycle contribute most to impact categories, but without revealing the position of each instances of the unit processes in the supply chain. This approach has been successfully applied to allow the application of site-dependant characterization factors (Mutel and Hellweg 2009).

2.2 Sequential method

2.2.1 Basic implementation

The sequential method is described as an alternative to solve the system of linear equations inherent to LCA, not

Table 1 Variables in the matrix representation

Name	Symbol	Content	Dimensions	Formula
Technology matrix	**A**	Consumption of intermediary products by other unit processes	$n\times n$	Normally supplied by the database
Intervention matrix	**B**	Emissions and natural resource consumption of unit processes	$m\times n$	Normally supplied by the database
Demand vector	f	Reference flow for the system	$n\times 1$	Set up by LCA practitioner
Scaling vector	s	Total quantity of intermediary products	$n\times 1$	$f=\mathbf{A}s \Leftrightarrow s=\mathbf{A}^{-1}f$
Inventory	g	Total emissions and natural resource consumption	$m\times 1$	$g=\mathbf{B}s$
Characterization matrix	**Q**	Characterization factors	$k\times m$	Normally supplied by impact method
Impact vector	h	Total score for each impact category	$k\times 1$	$h=\mathbf{Q}g$

simultaneously, but sequentially, hence its name (Heijungs and Suh 2002). It is a very instinctive way to present the global scope of life cycle assessment. It goes backward in the supply chain of a product or service, starting from the reference flows. When a final product is analyzed, the emissions of the site directly producing it will be quantified and stored. Then, the algorithm successively looks at its suppliers 1, 2 … n_0 and quantifies the output they need to produce to support the production of the reference flow. The emissions of these processes are quantified and stored, and those emissions are multiplied by characterization factors (CF) to calculate the impact score for all the categories considered. The same steps are repeated for supplier 1.1, 1.2 … 1.n_1 … 2.1, 2.2 … 2.n_2, etc. A more formal description of the sequential method (described in the context of structural path analysis, see Section 2.2.2) can be found in Peters and Hertwich (2006) or Defourny and Thorbecke (1984).

Theoretically, the supply chain could be extended infinitely. In practice, the impacts of processes very far in the supply chain are often insignificant compared to processes closer to the reference flows. When this is the case, those processes can be safely omitted, although one must keep in mind that the calculated total will inevitably be inexact.

Rigorously defining a criterion for cutoff based on distance from the initial reference flows is a delicate task. One might think that impacts are monotonously decreasing along the supply chain: The further the intermediate supplier relative to the functional unit, the smaller the impact. Power series have demonstrated that this is not the case. Figure 1 shows the distribution of supply chain impacts associated with the production of a concrete block, (life cycle inventories (LCI) data from the ecoinvent 2.0 database, impact scores evaluated with IMPACT 2002+). The stability of potential impacts in a specific branch of the supply chain over several tiers is

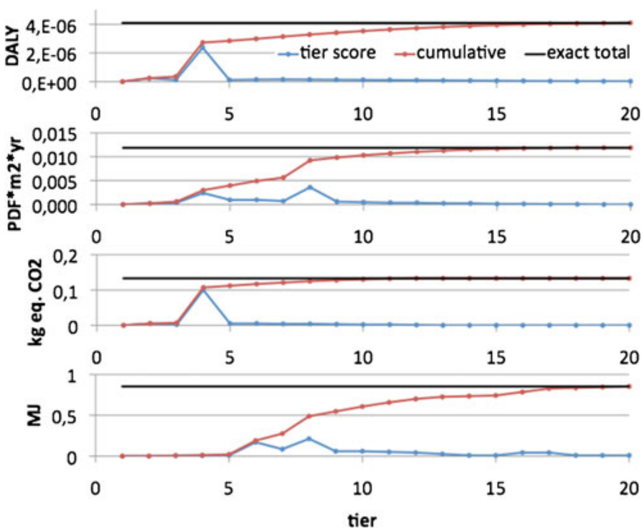

Fig. 1 Repartition of potential impacts along the supply chain for "Concrete block, at plant/DE," ecoinvent v2.0

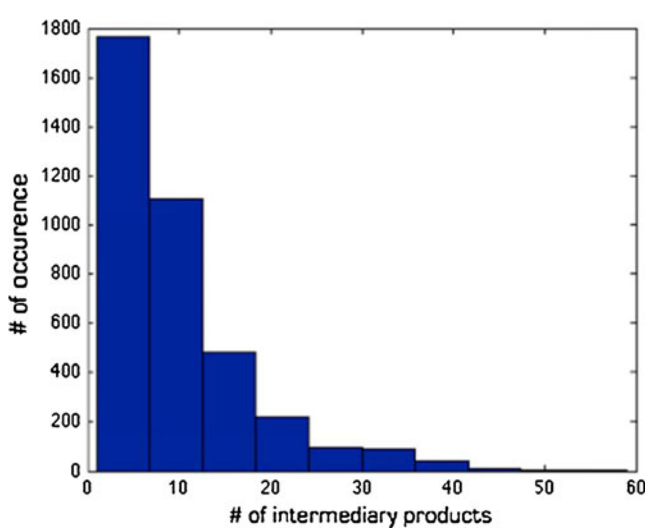

Fig. 2 Number of intermediary product flows per unit process in ecoinvent v2.0

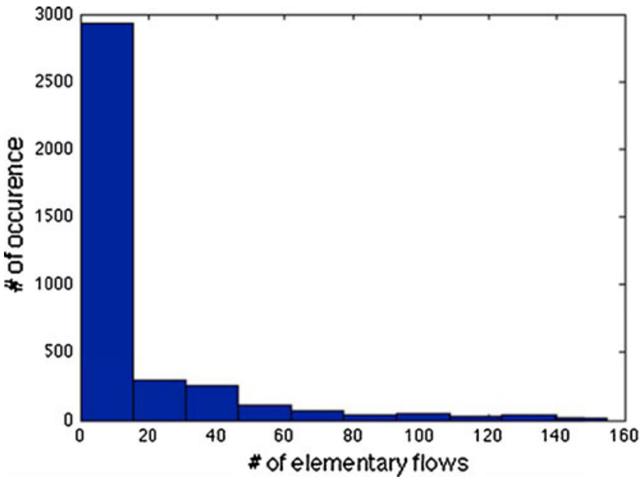

Fig. 3 Number of elementary flows per unit process in ecoinvent v2.0

not a reliable cutoff criterion, nor is a fixed number of tiers. In this particular example, for the Resource indicator, the contribution of each tier decreases steadily starting tier 8 and reaching at tier 14 an insignificant contribution compared to the cumulative impacts of the previous tiers. Cutting off after observing such a trend seems safe, but the cumulative curve shows that roughly 20 % of the impacts, calculated using matrix inversion, are still missing. The calculation has to be pushed further in the supply chain to capture the remaining impacts. The number of tiers necessary to capture a certain threshold of impacts (for example, 95 %) varies widely depending on the structure of the supply chain of the reference flows.

If one aims to minimize cutoff errors, the sequential method suffers from the exponential growth of the number of suppliers. In ecoinvent v2.0 (Frischknecht and Jungbluth 2007), each unit process requires, on average, 9.1 intermediary product flows, each of which is the output of another unit process. Also, each unit process has, on average, 21.6 elementary flows. For MIET, based on the USA 1996 input/output table (Suh and Huppes 2002), those numbers are 205.0 and 318.5, respectively. Figures 2 and 3 show a histogram of the distribution of those two numbers, noted $z_{\text{mean\_nnz}}$ and $B_{\text{mean\_nnz}}$, for ecoinvent v2.0.

The cumulative number of instances of unit processes and inventory entries can be estimated for product systems truncated after a number t of tiers:

$$\text{Cumulative number of instances of unit process entries in supply chain} \approx (z_{\text{mean\_nnz}})^t$$
$$\text{Cumulative number of inventory entries} \approx B_{\text{mean\_nnz}} \times (z_{\text{mean\_nnz}})^t$$

Table 2 shows those two values, calculated with ecoinvent v2.0 and MIET, for several values of t.

In a power series-based analysis of the ecoinvent 2.0 and the MIET databases using the IMPACT 2002+ methodology, it was shown that, respectively, 16 and 10 tiers are approximately necessary to capture 95 % of the impacts of a supply chain (Bourgault et al. 2009). Using this information, one can calculate the time and memory required to calculate an inventory with the sequential method. The sequential algorithm was coded in a Matlab environment, which can compute roughly 10,000 unit process entries per second on a regular laptop computer. The information on those 10,000 entries was stored in a variable of type "structure" and occupied 7 MB. If calculation had to be carried on to the 15th tier with ecoinvent database, only for unit process entries, computation time would be around 720 years, and the result would occupy 10,000 PB (peta=10^{15}).

Table 2 Cumulative number of entries produced by the sequential algorithm with ecoinvent v2.0 and MIET

t	Number of unit processes entries		Number of inventory entries	
	ecoinvent v2.0	MIET	ecoinvent v2.0	MIET
5	6.10e+04	3.62e+11	1.32e+06	1.15e+14
10	3.73e+09	1.31e+23	8.05e+10	4.18e+25
15	2.27e+14	4.75e+34	4.91e+15	1.52e+37
20	1.39e+19	1.72e+46	3.00e+20	5.49e+48

The usefulness of such a large quantity of information can be doubted. Although some meaningful contributors to final impact scores can be found up to the 16th tier, most branches of the supply chain have a negligible contribution. Computing the whole system would lead to an overwhelming number of unit process instances adding no further information about the impacts of the life cycle. Considering all these disadvantages, the sequential algorithm is not a viable way of computing the impacts of a product or a service with large databases if one is concerned with cutoff errors.

2.2.2 Structural path analysis and accumulative structural path analysis

The sequential method described in the previous section can be used to estimate life cycle inventories and associated impacts and to identify individual significant contributors.

A slight adaptation of the approach allows one to analyze sets of linked nodes, or "paths," that are of specific interest. This approach, termed structural path analysis, has been used with economic input–output (IO) models, which share many similarities with LCA regarding the structure of its quantitative information. Defourny and Thorbecke (1984) use SPA in the context of social accounting. Two poles are chosen and all the paths of influence between them are identified. The contribution of each path to the total influence is then calculated, revealing the ones of higher interest. In LCA terms, it is analogous to choosing two unit processes, listing all the supply chain links between them and calculating how the demand for one unit of the first unit process is transferred to the other unit processes via all the other unit processes of the database. Although of a certain practical interest, this is not typical of the type of information one aims to derive from LCA. Because the contribution of a path decreases with its length, Lenzen (2002), in his use of SPA to identify environmentally important input paths, fixes the maximum path length to five tiers. Figure 1 shows that this limit is somehow arbitrary and that it is certainly not adapted in the context of LCA. Peters and Hertwich (2006) apply SPA to determine how pollution is embodied in Norway's imports. Their approach is more typical of LCA, although they apply their cutoff criteria on a single metric (CO_2 emissions) and a maximal path length (eight tiers). This could lead to ignore significant paths. Being concerned with energy efficiency, Treloar's (1997) method uses a "threshold value" on embodied energy to reject insignificant paths.

Suh and Heijungs (2007) have formally described the procedure of using the system score at the end of a path. They make the distinction between SPA and accumulative structural path analysis (ASPA). In the former, the impacts of each individual occurrence of a unit process are revealed to find the most contributing one, using the unit score. In the later, an occurrence of a unit process is chosen in the supply chain; the impacts of this occurrence and all the other occurrences of unit processes between the chosen one and the reference flow are added, using the system score for the chosen one. The results are then displayed by path length. Suh (2004) uses SPA to show which occurrence of unit processes is the most contributing to the total impact. It enables him to draw conclusions about which unit process is an important direct and indirect source of pollution. This is an example of what would be called re-aggregation in the context of the systematic disaggregation. The proposed algorithm builds on the approach proposed by Suh by making it easy for the user to re-aggregate results around any poles deemed relevant in the context.

One of the key considerations is that the cutoff or relevance threshold should be based on multiple impact categories. Multiplying an inventory by characterization factors will drastically change the contribution of each elementary flow (and of each path), to the overall impact scores. The assumption that paths a certain lengths will not contribute significantly to the total impact is no longer safe in this context, especially, as seen below, for the Resource indicator.

2.3 Graphical representation allowed by both approaches

A fictional simplified system has been set up. The **Z** matrix of the system is shown at Fig. 4. Units in the matrix are "kg of product/service bought per kg of produced product/service." Figure 5a has been obtained using SimaPro, which uses matrix inversion, supposing an emission of 1 kg of CO_2 per kg of product supplied. This representation, known as "network," becomes hard to read in the presence of feedback loops, but it is the finest resolution that can be achieved when the linear system is solved with the matrix inversion algorithm. Figure 5b is the "tree" representation of the same system. The arrows in a tree representation always point in the same direction, and instances of a given unit process will appear as many times as necessary. It is much easier to read than the intricate network representation, but it implies the truncation at some arbitrary point in the supply chain (dashed arrows).

It should be emphasized that the notation used in this paper is different than the one of input/output analysis. It is more convenient in process-based LCA to use a technology matrix with unit processes normalized to an output of 1 unit. Also, full allocation should be preferred, leading to a square matrix with positive coefficients in the main diagonal (this point is further discussed in Section 3.1).

2.4 Adaptation of the sequential algorithm

The purpose of the calculation is to generate sufficient information to adequately interpret the results, relative to

		Demanding process				
		1	2	3	4	5
Supplying process	1	0	0	0	0	0
	2	0.5	0	0	0	0.5
	3	0.5	0.5	0	0.5	0
	4	0	0.5	0.5	0	0
	5	0	0	0.5	0.5	0

	Demanding process				
	1	2	3	4	5
kg of CO_2 eq. emissions/kg of product	1	1	1	1	1

Fig. 4 **Z** matrix (*top*) and **B** matrix (*bottom*) of a fictional system

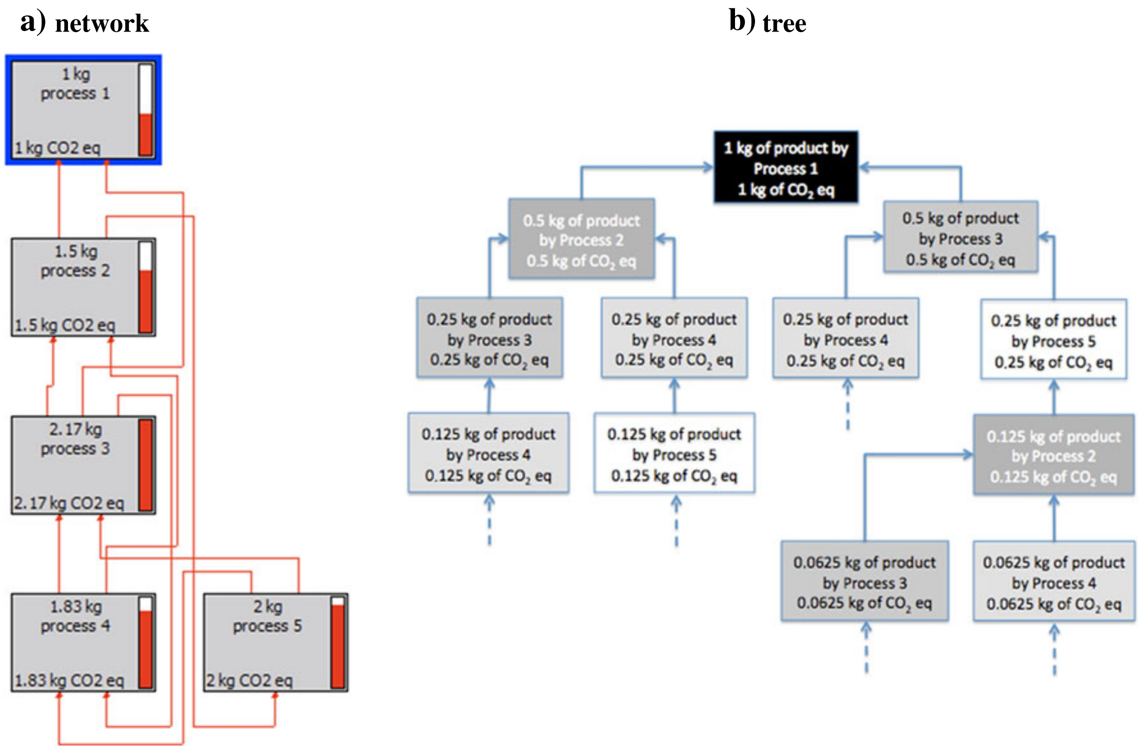

Fig. 5 Network and tree representation of the impact of 1 kg of output of "process 1"

the objectives of the study. Regardless of the objectives, finding the main contributors to each impact category is almost always on the list of things to do.

On the one hand, as exposed in Section 2.2, after an initial zone of apparent chaos, each new tier of the supply chain will bring an ever-decreasing contribution to impact scores. On the other hand, due to the exponential nature of the sequential method, those contributions will be spread over an ever-increasing number of contributors. Such a large body of insignificant impacts is not necessary to understand the key aspects of the life cycle of a product or a service. It is also not helpful because no action can be taken to tackle each of those sources of impacts: They are outside the sphere of influence of the sponsors of an LCA. Finally, the further down the supply chain a process is, the bigger is the uncertainty because there is no means to assess the representativeness of the data.

An algorithm producing a large amount of unhelpful and uncertain data after a long computation time is counter-productive. Yet, the method has merit because it allows one to access information necessary to build a tree representation of the supply chain. Consequently, the proposed adaptation of the sequential algorithm focuses on reducing the amount of unnecessary calculations, while retaining all the useful information. We have called it "systematic disaggregation." As will be seen, it shares many characteristics with SPA as described in, e.g., Peters and Hertwich (2006), but with certain adaptations to make the approach generalizable to multiple research questions, allow the calculation of exact results and to use multiple criteria, based on impact category scores, to define the resolution of the results.

The proposed solution is a hybrid between sequential algorithm and matrix inversion. The scheme of the calculation remains identical to the sequential algorithm, but the cutoff problem, mentioned earlier, is solved by comparing cumulative (system) impacts of a path to the impact of the whole system calculated by matrix inversion. If it is smaller than a certain criterion, the calculation is stopped for the specific branch, and the cumulative impacts of the upstream supply chain are stored. Table 3 shows the structure of the result file.

To implement the algorithm, the following steps are applied:

1. A pre-calculation is performed, avoiding the repetition of the same operations later in algorithm, hence saving computation time. The impact of every unit process is calculated for a unitary demand, using a final demand matrix equal to the identity matrix of dimension n. This is done for both cumulative, or "system," impact (using $\mathbf{QBA}^{-1}$) and unit process specific, or "unit," impact (using $\mathbf{QB}$). This results in two $k \times n$ matrices containing the system and unit scores for each of the k impact categories, for each of the n unit process. This pre-calculation is specific to the technology matrix used, but not to the product system analyzed, so it can be reused for several analysis

Table 3 Structure of the output table

Column of the result table	Comment
(a) Instance ID number	Unique ID number for each line, allowing the differentiation of instances of the same unit process in the product system
(b) Instance ID number of the parent branch	Useful in the construction of the tree
(c) Demand	Scaling factor for each instance of a unit process, e.g., how much of the intermediate flow is required, along with the unit (kWh, kg, etc.)
(d) Unit process ID	The name and number of the unit process, as described in the database
(e.1) Category 1 unit process score (e.2) Category 2 unit process score	The scaled impact scores associated with each instance of the unit process
(e.3) Category *k* unit process score	
(f.1) Category 1 system process score	The scaled impact scores caused by each instance of a unit process and of its supply chain
(f.2) Category 2 system process score	
(f.3) Category *k* system process score	
(g) Aggregation flag	Flag indicating whether the system score of the instance divided by the total score is smaller than the disaggregation criterion, for all categories. If the score is larger than the disaggregation criterion for all impact categories, it takes the value "disaggregated," and the instance ID number will appear in later lines in column b. If smaller, then it takes the value "aggregated"
(h) Path length	Number indicating how many tiers are found between this instance and the reference flow
(i.1) Level 1 unit process ID (i.2) Level 2 unit process ID (i.3) Level [level of supply chain] unit process ID	Information allowing the visualization of the path without having to refer to a sequence of parent branch number scattered over the entire table
(j) Country tag	Complementary information about the nature of the unit process or the emission, allowing the creation of customized re-aggregation criteria in the interpretation phase
(k) Infrastructure process tag	
(l) Process categories (i.e. economic sectors) (m) Other complementary information tag, customized by the practitioner	

as long as it can be described using the same technology matrix (e.g., using the same LCI database).

2. A system is built by a practitioner with the unit processes available in the technology matrix. This is done by listing in the final demand vector (**f**) the quantified reference flows.
3. Since this system is a linear combination of elements of the technology matrix, the system-wide score (i.e., the life cycle impacts) can be quantified using the pre-calculated system scores performed at step 1.
4. A disaggregation criterion is chosen. This is not the same as a cutoff criterion, the latter implying that the results do not cover 100 % of the impacts. As the life cycle impacts are calculated using matrix inversion, the disaggregation criterion does not affect the final result. Rather, it only changes the amount of information that will be revealed in a disaggregated fashion. The smaller the disaggregation criterion, the more detail will be available. It is difficult to choose the appropriate disaggregation criterion before performing any calculation. If the algorithm does not reveal the necessary information, it might be necessary to choose a smaller criterion and perform the calculation again.

5. The algorithm starts with one of the intermediary product flows present in the final demand vector. A unique branch ID is created (see Table 3, element a), the demand and the name of the unit process are stored (see Table 3, elements c and d). The unit and system scores have been calculated for a unitary demand at step 1 (see Table 3, elements e and f). Calculating the unit and system scores for this branch is simply a matter of scaling them according to the demand.
6. For each impact category, the system score is divided by the total score of the whole reference flow. The result represents the relative contribution of the impacts of this unit process and its entire supply chain. If this contribution is smaller than the disaggregation criterion for each category, no further details should be computed for this branch and the aggregation flag is set to "aggregated" (see Table 3, element g).
7. The path length and the path between the branch and the functional unit are stored (see Table 3, elements g and i). Since the parent branch number is stored, it is easy to retrieve this information. Additional tags can also be stored (see Table 3, elements j through m).
8. Steps 5 through 7 are repeated for each unit process necessary to supply level 1 (reference flows). The algorithm then scans through every branch of level 2. The branches that have been identified as aggregated will not be submitted to any further calculation. To speed up the calculation, the system score of the branch to be evaluated (previously calculated for an arbitrary unit demand in step 1) is scaled by the corresponding a_{ij} of the technology matrix and the demand of the parent branch. The algorithm presented by Peters and Hertwich (2006) recalculates the score from scratch. We believe this is why they report a calculation time of several minutes to 2 h. The approach proposed here is avoiding the unnecessary duplication of calculation. The algorithm will scan each new level of the supply chain until every instance generated receives the tag "aggregated." The calculation is then completed.

3 Results and discussion

3.1 Condition for convergence and analysis

Algorithm based on the sequential method has similar conditions for convergence as those of the power series algorithm (Suh and Heijungs 2007). For the power series to converge, elements in the main diagonal of the technology matrix have to lie in the interval [1, 2[. For algorithm based on the sequential method, the condition is identical. This is because the **Z** matrix, defined as $\mathbf{I}-\mathbf{A}$, is used to calculate the supply for each unit process. The ecoinvent database has a main diagonal equal to 1, so no unit process ever supplies itself. This is not the case in an input/output table. Due to intra-sector sales (for example, cattle food is sold to a bovine farm, both part of the same sector), the main diagonal of the **Z** matrix might not be null. In the MIET database, the diagonal contains value ranging from 1 to 1,3378. In the network representation, this situation translates as an arrow pointing back toward its sector of origin. In a tree representation, the process will be repeated over and over again with an ever-decreasing demand, as in a fractal. This explains why the **Z** diagonal has to range between 0 and 1. If it would be equal or greater than 1, a stationary or ever-increasing demand of the same process would propagate through the tree and the algorithm would never converge. It is possible to apply the proposed approach to IO-LCA, but columns of the technology matrix have to be rescaled to fulfill the main diagonal condition. Since the computational structure of hybrid LCA is also similar, if not identical, to process LCA, systematic disaggregation can also be to a database constructed with this methodology.

For the power series to converge, it is not necessary that each element outside the main diagonal of the **A** matrix be negative, i.e., that processes should be fully allocated (Peters 2007). In other words, multi-product unit processes are allowed, as long as the technology matrix remains square. However, in the case of incomplete allocation, the term-by-term contribution will include the avoided (i.e., negative) production of other unit processes. This make results of the power series approach more difficult to interpret. This is not the case for the algorithm based on the sequential method. Each instance of avoided impact can be traced back to the original multi-product unit process. Each possible reference flow of the ecoinvent 2.0 and the MIET database was tested with the systematic disaggregation algorithm and the power series algorithm. Neither of the databases is fully allocated, and the calculation for every reference flow converged to the score calculated with the matrix inversion.

Finally, because of the multiplication of the **Z** matrix by itself, an additional criterion is imposed on the power series algorithm. The spectral radius of the **Z** matrix (the modulus of the largest eigenvalues) should be smaller than unity. This is not a necessary condition for algorithm based on a sequential approach to converge.

3.2 Characteristics and originality of the output

The result of the algorithm does not look like the more familiar scaling vector, inventory, and scores per category. The result is a table that allows access to the full details of the supply chain tree and can be consulted in its raw form. It can also be subjected to re-aggregation and filtering to produce easy-to-understand figures.

Systematic disaggregation reports both unit impact (caused by the direct emissions of the instance of the unit process) and system impact (caused by direct emissions of the unit process and the rest of its upstream supply chain). This characteristic has several advantages. First, it is easy to check if the

calculation has been performed correctly: The system scores of the instances of unit process that were aggregated and the unit scores of the ones that were disaggregated should sum up to the initially calculated system score of the whole product system, for each category. This is why we talk about "disaggregation criterion" instead of "cutoff criterion." The decision not to dig deeper in the supply chain leads not to a loss of impact but only a loss of detail on which unit process is causing the impacts. Also, a practitioner might be interested in looking both unit and system scores. It might facilitate understanding, calculation or reporting. When a unit score is high, it points out to a potentially actionable item: It can guide the decision maker (supply chain management decision, eco-design, etc.) and the practitioner (i.e., in primary data collection, see example in Section 3.3).

In the short version of the output table (called unit process output table), a line gathers the information on an occurrence of a unit process. This means that the same unit process can appear more than once. This will occur in the presence of feedback loops, or if the same unit process is part of the supply chain of more than one reference flow or intermediary unit processes. This is the main difference between the systematic disaggregation algorithm and the matrix inversion algorithm. In the latter, all the instances of a unit process are automatically aggregated in the matrix inversion operation. They cannot be dissociated. Details on which elementary flows are emitted and how they contribute to the score of the occurrence is not shown.

In its long version (called inventory output table), each line represents a single elementary flow with its associated impact scores. This output is obtained by breaking down the contribution of each elementary flow to the score of each occurrence of unit processes. Which occurrence of unit process is responsible for the elementary flow and the path between the elementary flow and the reference flow are also shown, in addition to all the information present in the unit process output.

The output of the algorithm takes more memory than a traditional inventory list but is still easily managed by a personal computer. The size of the output depends on the disaggregation criterion. A smaller criterion will lead to a larger output because the algorithm will have to push further in the supply chain. Figure 6 shows the number of lines of the unit process output as a function of the disaggregation criterion for sample ecoinvent processes. Figure 7 shows the "depth" of the tree, or how many rounds of the supply chain are covered, before the algorithm stops. Figure 8 displays the number of revealed unit processes for a particular round of the supply chain (the "width" of the tree) as a function of the tree depth for different disaggregation criterion. Finally, Fig. 9 shows what proportion of the impacts are shown as disaggregated. This quantity is not equal to the disaggregation criterion. If the disaggregation criterion is 1 %, each instance of a unit process contributing to less than 1 % will be kept aggregated. If, for example, two unit processes contribute to 0.99 % of the score, they will both be kept aggregated and their sum will be higher than 1 %.

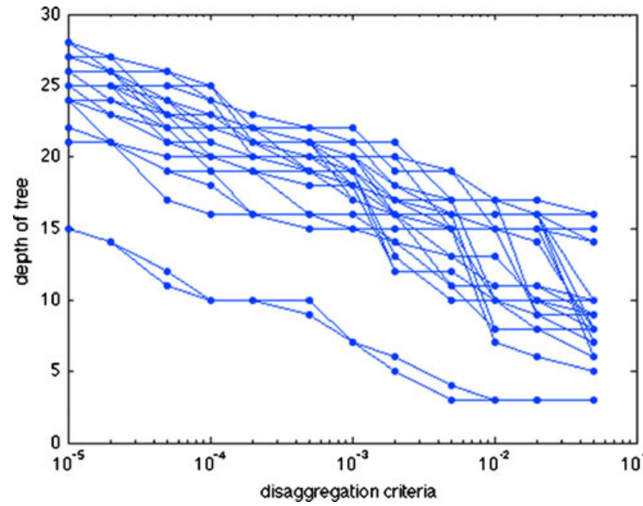

Fig. 6 Tree depth as a function of the disaggregation criterion for a random sample of 26 processes of the ecoinvent database

The difference between ASPA as presented in Suh and Heijungs (2007) and the current approach is subtle. In ASPA, the impacts of every contributing occurrence of unit processing a path are summed, from the reference flow to the chosen "end of path" occurrence of unit process. This does not allow for a detailed analysis of every contributor on the path. The analysis has to be complemented by SPA and both outputs have to be interpreted in conjunction of each other. Systematic disaggregation keeps separated the contribution of each occurrence on paths, until the system score contributions become smaller than the disaggregation criteria. The contribution can be quickly re-aggregated to obtain the same results as ASPA using filters and summing the right columns of the output.

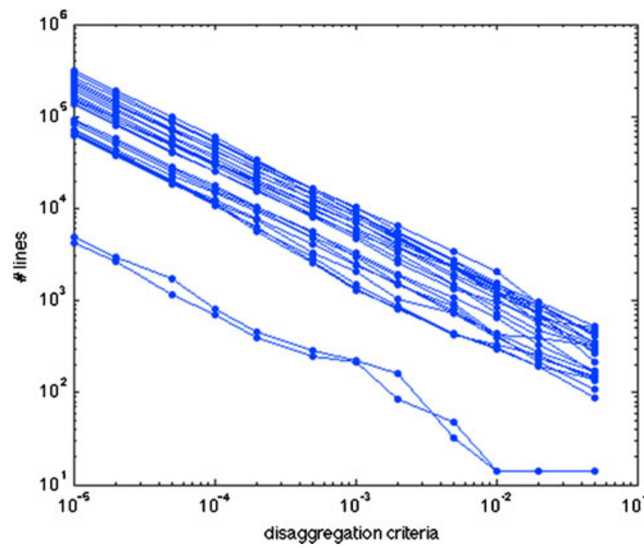

Fig. 7 Length of the unit process output table as a function of the disaggregation criterion for a sample of 26 processes of the ecoinvent database

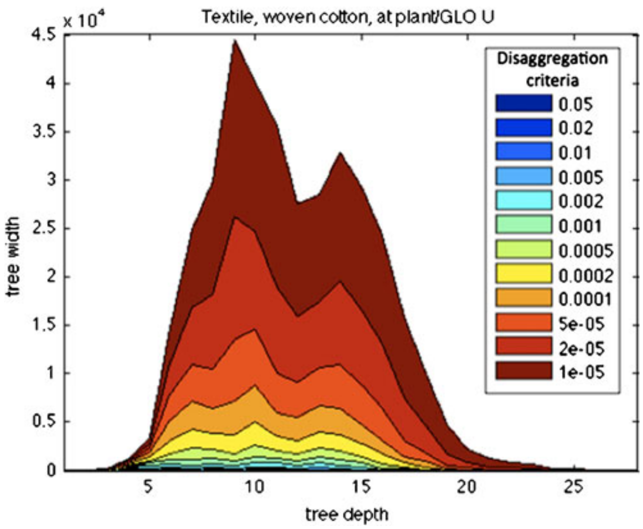

Fig. 8 Width of the unit process output tree varying with tree depth and disaggregation criterion for a randomly chosen ecoinvent unit process

3.3 Application: selective attribution of site-dependant characterization factors

Some impact categories display a high spatial variability. The major contributors to those impacts are often in the background of the model, i.e., they are represented by a generic unit process. If the matrix inversion is used, the major contributions are arbitrarily added to a very large number of contributions from other branches of the supply chain. Those contributions might individually be very small, but because of their very high number, they will add up to a meaningful value compared to the major contributor. It is impossible, and also not necessary, to gather spatial information on the large number of small contributors. It can, however, be assumed that the pressure will keep building to do so for major contributors, since more and more spatialized CF will be available in the near-future. It is a task that has been retrospectively done for the acidification impact category (Bellekom et al. 2006), and it is deemed a reasonable demand on practitioners.

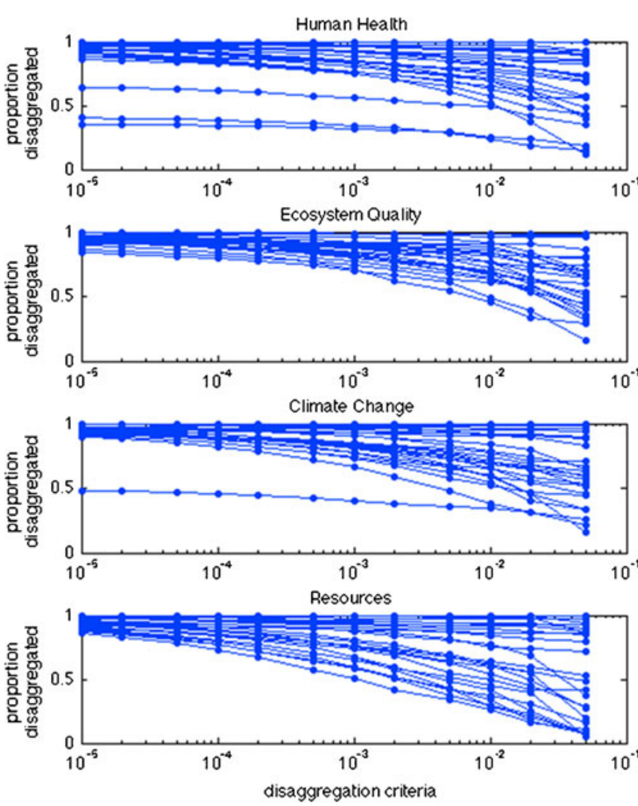

Fig. 9 Proportion of impacts disaggregated for a sample of 26 processes of the ecoinvent database

After identifying that a unit process is indeed a major contributor, the first task is to disaggregate it in its different occurrences. The systematic disaggregation algorithm allows doing so. Inspection of the paths of each contributor is displayed, and the potential candidates for attribution of site-dependant CF are easily identified. At that point, a new column can be added to the output, where the practitioner will proceed to tag each candidate for site-dependant CF attribution with a number. Using pivot table, the scores can be re-aggregated according to the tag in the newly filled column. Those left unidentified will be added together by default, and those identified with a specific site number will appear in separate entries. Replacing their site-generic scores by site-dependant scores becomes a straightforward task.

The previous example focuses on selective attribution of CF, but this is a specific application of a more general functionality of the systematic disaggregation algorithm: replacement of values (CF, emission rates, intermediary product demand) affecting only the portion of the supply chain that makes sense. Showing the disaggregated results allows the practitioner to prioritize the primary data collection. By making no assumption and including all potentially useful information in the output, the algorithm creates almost infinite flexibility of manipulation, adapting itself to the context of each study.

3.4 Application: re-aggregation around scopes 1, 2, and 3

Let us assume the goal and scope of an LCA include the separation of emissions according to scopes 1, 2, and 3 (as defined by the Greenhouse Gas Protocol) of the production of 1 kg of paper by an integrated paper mill. The company running these activities does not buy wood harvested by another company but carries on all aspects of the production, from the management of the forest to the final product. According to the GHG Protocol (World Business Council for Sustainable Development 2004), scope 1 emissions occur in facilities or devices (such as vehicles) owned or controlled by the company. Scope 2 emissions are the one

caused by the production of electricity bought and consumed by the company. Scope 3 emissions occur in facilities that supply the company but are not owned by it or under its control. Emissions from a wide variety of unit processes should be included in the scope 1, and the electricity bought by all those activities should count as scope 2. However, electricity (potentially from the same power plants) consumed by the company's suppliers would fall under scope 3 and yet be represented by the same unit process in the LCI database. It becomes clear that highly aggregated results as produced by the matrix inversion are not suited to make the necessary distinctions.

Making a demand of 1 kg of "Paper, newsprint, at plant/CH" in ecoinvent will lead to impact scores of 5.84×10^{-7} DALY, 1.757×10^{-1} PDF m^2 year, 8.035×10^{-1} kg$_{eq}$ CO_2, and 2.077×10^{1} MJ, when the impacts are evaluated with IMPACT 2002+ endpoint characterization factors. The unit process output generated with a disaggregation criterion of 0.5 % leads to 4,269 entries. Thirty of those belong to scope 1, including "Paper, newsprint, at plant/CH" itself and some occurrences of "Diesel, burned in building machine/GLO," "Operation, lorry >16 t, fleet average/RER," and "Power sawing, without catalytic converter/RER." A column called "scope" is added to the output, and the mentioned occurrences are tagged with a "1" in this column. Although this identification has to be done manually, a clever and systematic use of the filters in Excel makes it relatively easy and quick process, with minimal risk of omission.

The scaling vector produced by the matrix inversion informs us that a total of 2.198×10^{-2} MJ of "Diesel, burned in building machine/GLO" is necessary in the whole supply chain. The disaggregated unit processes output reveals 33 instances of this unit process. Among them, five have been tagged as part of scope 1. They amount to 9.987×10^{-3} MJ, or 45.4 % of the total demand of this unit process. The rest of the demand occurs elsewhere in the supply chain, for example: "Paper, newsprint, at plant/CH" → "Sodium dithionite, anhydrous, at plant/RER" → "Zinc, primary, at regional storage/RER" → "Diesel, burned in building machine/GLO" which is clearly part of scope 3.

Identifying instances of unit processes part of scope 2 is just as straightforward. Filtering to show only energy production unit processes, the instances directly linked to an electricity demand originating from activities within scope 1 will be attributed a "2" in their scope column. One hundred seven of them can be tagged as such in the present example. The remaining 4,132 instances of unit processes, neither part of scope 1 nor 2, fall by default in the scope 3.

Using Pivot Tables under Excel will aggregate the scores around the newly created tags scopes 1, 2, and 3. A second dimension can be added to create a histogram as in Fig. 10. The path length can be used as a second re-aggregation criterion.

For the re-aggregation of the scores, it is necessary to take the unit score of the instances identified as part of scores 1 and 2. Otherwise, impacts of activities occurring outside those activities will be taken into account. Then, the impact of scope 3 is the difference between the total and what is part of scopes 1 and 2.

3.5 Application: estimation of the number of relevant production sites

As mentioned earlier, systematic disaggregation is also adapted to answer research questions. We show here how systematic disaggregation is used to approach a question about the number of production sites appearing in the model.

Application of spatialized characterization factors requires that the actual location of an emission is known. In the absence of this information, a site-generic characterization factor should be used (Udo de Haes et al. 1999). A relevant question has since been asked: from how many sites do life cycle emissions originate (Norris 2002)? According to Norris, with an input–output database, pushing the calculation to the 6th tier will capture 95 % of the criteria air pollutants and toxic releases for 75 % of the commodities, which means approximately 2.8×10^{15} instances of unit processes. The number of actual *sites* is potentially much smaller because the same sites can occur at several instances in a supply chain. Coming back to Fig. 5b, the different instances of the same unit process could be representing the same production site, or different ones with a similar output, but it is impossible to determine because the information comes from a generic database.

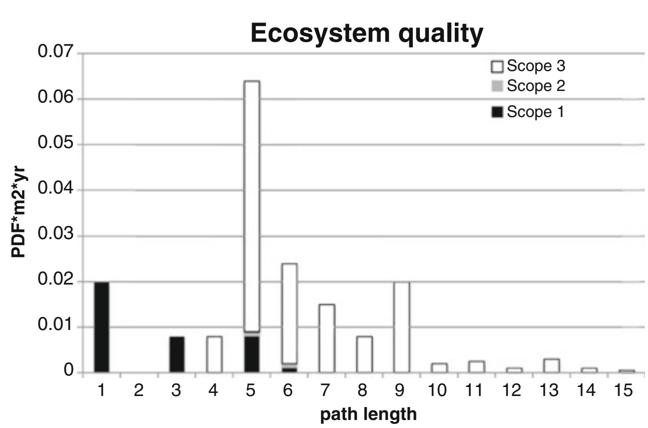

Fig. 10 Distribution of impact scores by path length of the supply chain and scope for Ecosystem quality impact category

As mentioned in Section 3.3, this is an important question: Knowing the precise coordinates is necessary to choose the right spatialized CF. Not knowing means introducing either a bias or uncertainty in the result. Using a streamlined simplification of the database, Norris shows that the 1,000 process instances with the most important contribution will sum up to less than 50 % of the emissions for a typical product.

The systematic disaggregation algorithm provides a more precise answer to the question. Specifying criteria of 5, 2, 1, 0.5, and 0.1 %, the disaggregation is applied to every unit process of the ecoinvent database. For each criterion and process, the proportion of impact generated by disaggregated instances of unit processes is plotted against the number of those disaggregated instances. If the 4,000 curves are plotted at once, the resolution does not allow for a lot of observations. But plotting a random selection of 10 % of the curves as in Fig. 11 shows that some zones are denser than others. Taking a "cut" at 400 on the *x*-axis creates a histogram as in Fig. 12. The former figure tells us that as a general rule, the effect of revealing more instances of processes in the supply chain is fairly independent of the initial reference flow. Also, after a certain point that appears to be around 300 instances, revealing more instances reveals less and less impacts because of the high dispersion of the impacts over a large number of instances in higher tiers. The histogram shows that for a specific number of disaggregated instances, the proportion of impacts revealed varies wildly: No general rule can be established. It is our opinion that specifying an average, in the presence of such a large variance, would be misleading.

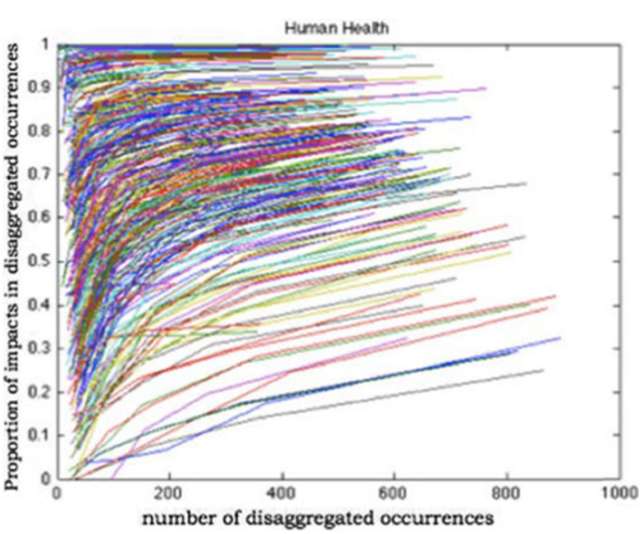

Fig. 11 Impact in disaggregated instances of unit processes

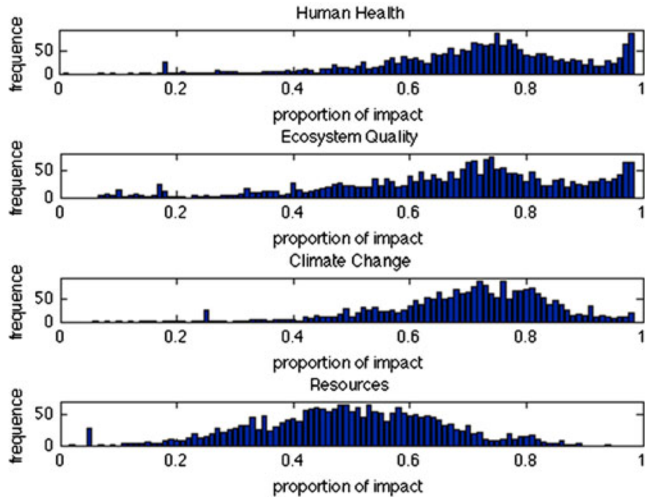

Fig. 12 Distribution of impact in disaggregated unit processes for 400 instances of unit processes

4 Conclusions

The proposed algorithm is a hybrid between the sequential algorithm and matrix inversion algorithm. It allows the access to details of the supply chain provided by the sequential algorithm and precision of the matrix inversion algorithm. Compared to previously published algorithm, it fixes a threshold (disaggregation criterion) for significant paths based on multiple impact scores, ensuring the display of all the paths significantly contributing to the total scores. The output of the algorithm can be easily interpreted by simple inspection, or with data mining tools like pivot table and graphs, or database software. With a regular personal computer, the computation is carried on within seconds with some of the largest disaggregated database available so far (ecoinvent and MIET) and for very small disaggregation criteria (10^{-5})

The result structure facilitates the interpretation of the life cycle of a product or a service. With minimal adaptation and a bit of imagination, the algorithm can also be used to answer research questions, such as how the uncertainty propagates through the supply chain. For example, since the algorithm displays all the significant paths, which are a minority, Monte Carlo simulation could be applied only to those paths. This would greatly speed the computation.

Acknowledgments The authors acknowledge the financial support of the industrial partners in the International Chair in Life Cycle Assessment (a research unit of CIRAIG): ArcelorMittal, Bell Canada, Cascades, Eco Entreprises Québec, RECYC-QUÉBEC, Groupe EDF, Gaz de France, Hydro-Québec, Johnson & Johnson, Mouvement des caisses Desjardins, Rio Tinto Alcan, RONA, SAQ, Total, and Veolia Environment.

References

Bellekom S, Potting J et al (2006) Feasibility of applying site-dependent impact assessment of acidification in LCA. Int J Life Cycle Assess 11(6):417–424

Bourgault G, Lesage P et al (2009) Enhancing LCA interpretation with an algorithm preserving total disaggregation. In: SETAC Europe: 19th Annual Meeting, Göteborg, Sweden

Defourny J, Thorbecke E (1984) Structural path analysis and multiplier decomposition within a social accounting matrix framework. Econ J 94(373):111–136

Frischknecht R, Jungbluth N (2007) ecoinvent report No.1., pp 1–77

Heijungs R, Suh S (2002) The computational structure of life cycle assessment. Kluwer Academic, Leiden

Lenzen M (2002) A guide for compiling inventories in hybrid life-cycle assessments: some Australian results. J Clean Prod 10(6):545–572

Mutel CL, Hellweg S (2009) Regionalized life cycle assessment: computational methodology and application to inventory databases. ES&T 43(15):5797–5803

Norris G (2002) Life cycle emission distributions within the economy: Implications for life cycle impact assessment. Risk Anal 22(5):919–930

Peters GP (2007) Efficient algorithms for life cycle assessment, input–output analysis, and Monte-Carlo analysis. Int J Life Cycle Assess 12(6):373–380

Peters GP, Hertwich EG (2006) Structural analysis of international trade: environmental impacts of Norway. Econ Sys Res 18(2):155–181

Suh S (2004) Functions, commodities and environmental impacts in an ecological–economic model. Ecol Econ 48(4):451–467

Suh S, Heijungs R (2007) Power series expansion and structural analysis for life cycle assessment. Int J Life Cycle Assess 12 (6):381–390

Suh S, Huppes G (2002) Missing inventory estimation tool using extended input–output analysis. Int J Life Cycle Assess 7(3):134–140

Treloar GJ (1997) Extracting embodied energy paths from input–output tables: towards an input–output-based hybrid energy analysis method. Econ Sys Res 9(4):375–391

Udo de Haes H, Jolliet O et al (1999) Best available practice regarding impact categories and category indicators in life cycle impact assessment. Int J Life Cycle Assess 4(2):66–74

World Business Council for Sustainable Development (2004) The greenhouse gas protocol. World Business Council for Sustainable Development, Geneva, pp 1–116